

CLAIMS

We claim:

- 1 1. A software tool, comprising:
2 logic configured to enable a child process that inherits the address space of an
3 instrumented parent process to execute as if the child process was generated from an
4 unaltered version of the address space when the child process inherits the altered
5 address space of the parent process.
- 1 2. The software tool of claim 1, further comprising:
2 logic configured to enable execution of the altered address space after the child
3 process terminates.
- 1 3. The software tool of claim 2, wherein the logic configured to enable
2 execution of the altered address space further comprises:
3 a breakpoint store configured to receive a list of breakpoints inserted into the
4 address space during instrumentation of the parent process; and
5 a backpatch engine configured to receive the list, the backpatch engine further
6 configured to reinsert the breakpoints in the address space.
- 1 4. The software tool of claim 3, wherein the backpatch engine is
2 responsive to an event indicative of termination of the child process.
- 1 5. The software tool of claim 1, wherein the logic configured to enable a
2 child process that inherits the address space of a parent process to execute further
3 comprises:
4 a process image store configured to receive an original instruction bundle from
5 the address space of the parent process when the original instruction bundle is altered
6 with a breakpoint during binary instrumentation of the parent process; and
7 a backpatch engine configured to receive the original instruction bundle and
8 replace the breakpoint with the original instruction bundle in the address space.

1 6. The software tool of claim 5, wherein the backpatch engine is
2 responsive to an event indicative of a breakpoint encountered during execution of the
3 child process.

1 7. A method for processing breakpoint events in a child process created
2 from a parent process, wherein the parent process is instrumented by a software tool,
3 the method comprising:
4 storing uninstrumented parent process code replaced by each occurrence of a
5 breakpoint inserted into the address space during instrumentation of the parent
6 process;
7 monitoring execution of a child process created by the parent process for an
8 initial breakpoint in the address space;
9 suspending execution of the child process in response to the initial breakpoint;
10 replacing each occurrence of a breakpoint in the address space with the
11 uninstrumented parent process code; and
12 resuming execution of the child process.

1 8. The method of claim 7, wherein storing comprises retaining a copy of
2 the instructions replaced by the breakpoint in a process image store.

1 9. The method of claim 7, wherein monitoring comprises executing a
2 process monitor configured to respond to trace events generated by the child process.

1 10. A method for run-time measuring a parent process instrumented by a
2 software tool, where the parent process includes a fork instruction, the method
3 comprising:
4 storing each occurrence of a breakpoint located in an address space associated
5 with a parent process during instrumentation of the parent process;
6 monitoring execution of the parent process for an indication that the parent
7 process is about to resume execution after the termination of a child process generated
8 in response to a fork instruction, wherein the address space has been altered;
9 suspending execution of the parent process in response to the indication that
10 the parent process is about to resume after termination of a child process generated in
11 response to the fork instruction;
12 restoring each breakpoint located in the address space during instrumentation
13 of the parent process to the address space; and
14 resuming execution of the parent process.

1 11. The method of claim 10, wherein storing comprises retaining a list
2 including the breakpoint and an associated address in a breakpoint store.

1 12. The method of claim 11, wherein restoring further comprises applying
2 the list.

1 13. The method of claim 12, wherein applying comprises inserting the
2 breakpoint at the associated address within the address space.

1 14. The method of claim 10, wherein monitoring comprises executing a
2 process monitor configured to respond to trace events generated by the parent process.

1 15. A computer-readable medium, comprising:
2 logic configured to enable a child process that inherits an instrumented address
3 space of an instrumented parent process to execute as if the child process was
4 generated from an uninstrumented address space; and
5 logic configured to enable execution of the instrumented parent process after
6 the child process terminates.

1 16. The computer-readable medium of claim 15, wherein the logic
2 configured to enable execution of the instrumented parent process further comprises:
3 logic configured to receive a list of breakpoints inserted into the address space
4 during instrumentation of the parent process; and
5 logic configured to reinsert the breakpoints in the address space.

1 17. The computer-readable medium of claim 16, wherein the logic
2 configured to reinsert the breakpoints is invoked in response to an indication that the
3 child process has terminated.

1 18. The computer-readable medium of claim 15, wherein the logic
2 configured enable a child process to execute further comprises:
3 logic configured to receive an original instruction bundle from the address
4 space of the parent process when the address space is altered by inserting a breakpoint
5 during binary instrumentation of the parent process; and
6 logic configured to replace the breakpoint with the original instruction bundle
7 in the address space.

1 19. The computer-readable medium of claim 18, wherein the logic
2 configured to replace the breakpoint with the original instruction bundle is responsive
3 to an indication that child process encountered the breakpoint during execution.